SENSOR-BASED UNDERWATER TRACKING OF MARINE OBJECT

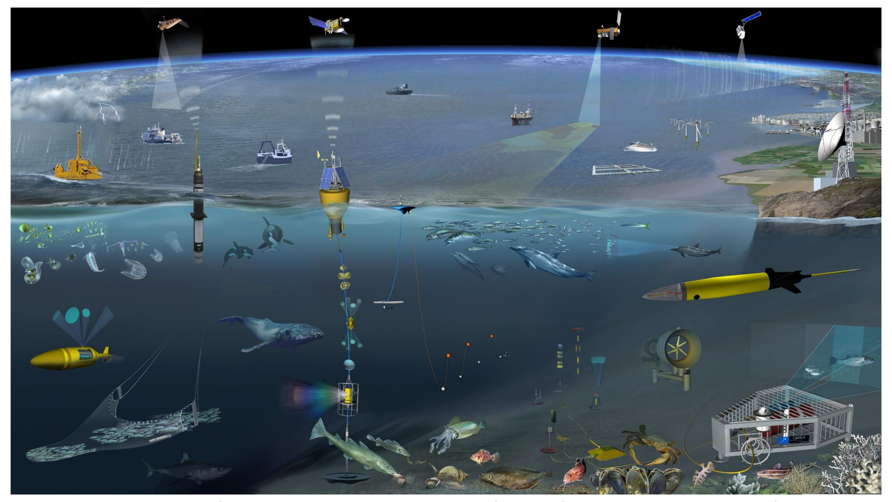
Emil Ramovic (s194149), Gergely Pap-Takács (s222677), Jonas Bolduan (s212704), Lucas M. Sandby (s223093)

Autonomous Marine Robotics - DTU, Technical University of Denmark

BACKGROUND

- growing interest in ocean exploration and exploitation for scientific & commercial purposes
- domains: scientific, industrial, transport, human interaction
- oceanic environment motivates **sensor fusion** (i.e. optical & acoustic) to mitigate weaknesses & errors
- reliable underwater localization and navigation current technological challenge

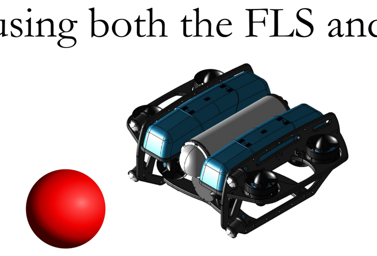
Application example: within current marine robotic trends: **Robotic network** interaction between carrier and its one or more passenger robots (heterogeneous robots achieving more complex missions)



Depiction of ocean observation system (taken from Whitt et al. 2020)

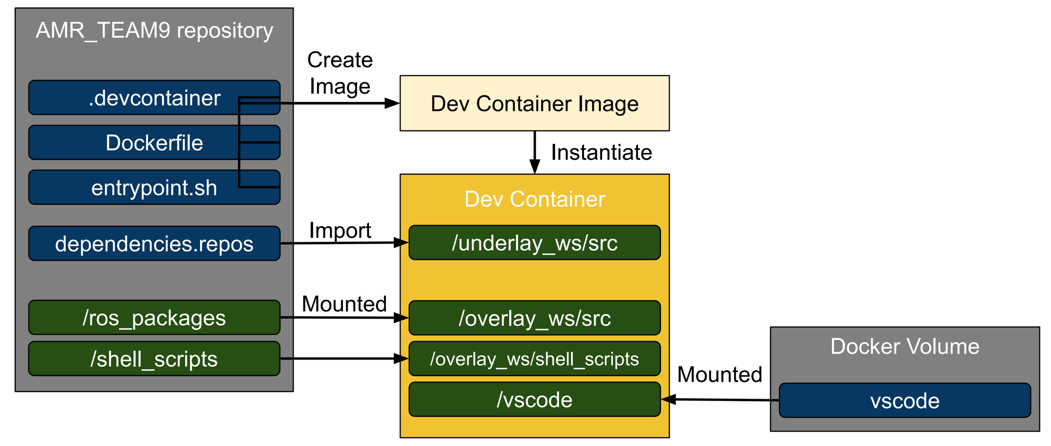
Project: Red sphere as object of interest (OOI), that the ROV tries to detect & follow using both the FLS and the Stereo Camera

MVP: Track linear movement with camera

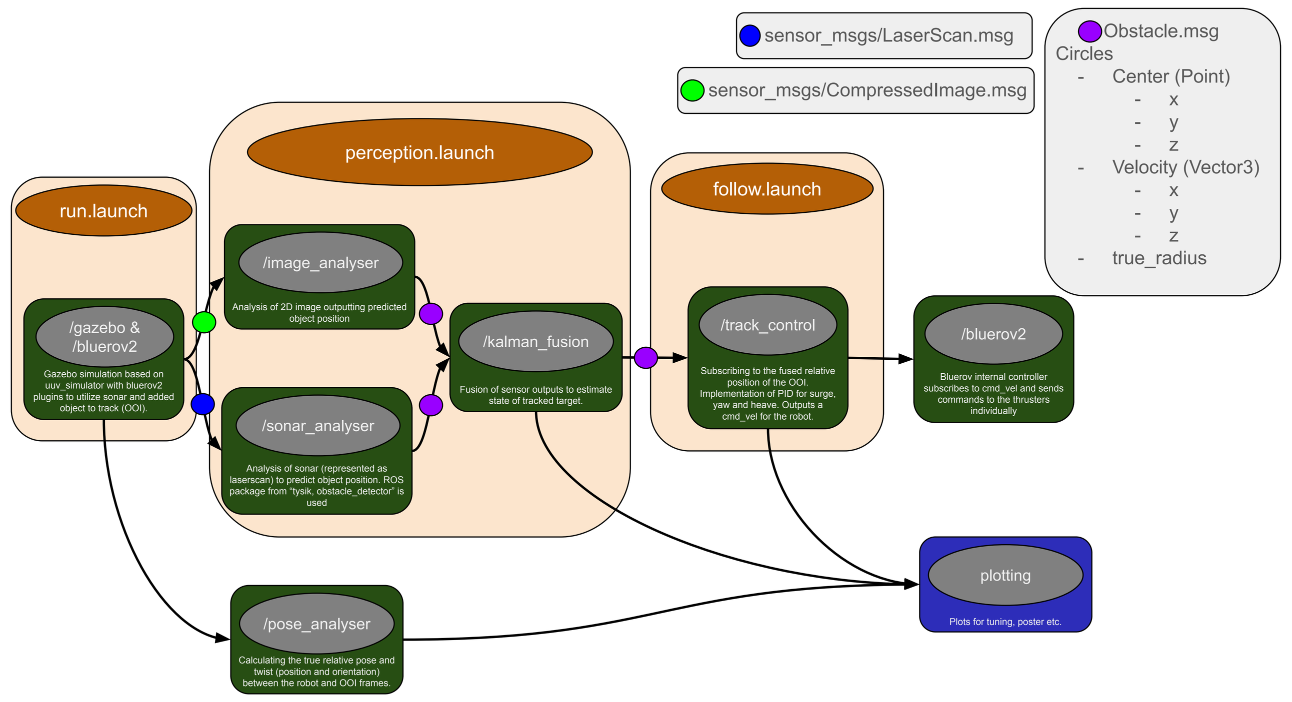


DEVELOPMENT ENVIRONMENT

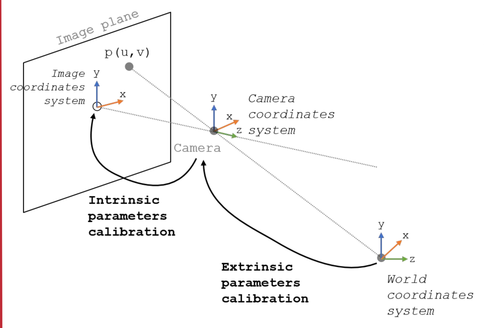
- Dev container with underlay containing base packages used (simulator etc.) and overlay with custom packages
- ROS package for amr-project containing all the custom code for this project



ROS ARCHITECTURE & CUSTOM MSG



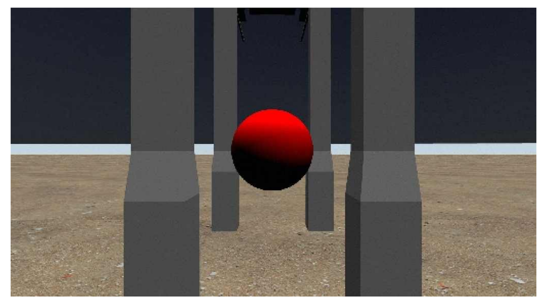
PERCEPTION: CAMERA



How camera captures the world

Image object detection:

- BlueROV2 has multiple cameras
- In the simulation we used the RGB camera
- We use the standard bridge setup and standard lighting conditions
- Detection of the red ball is heavily affected by these choices



Camera picture

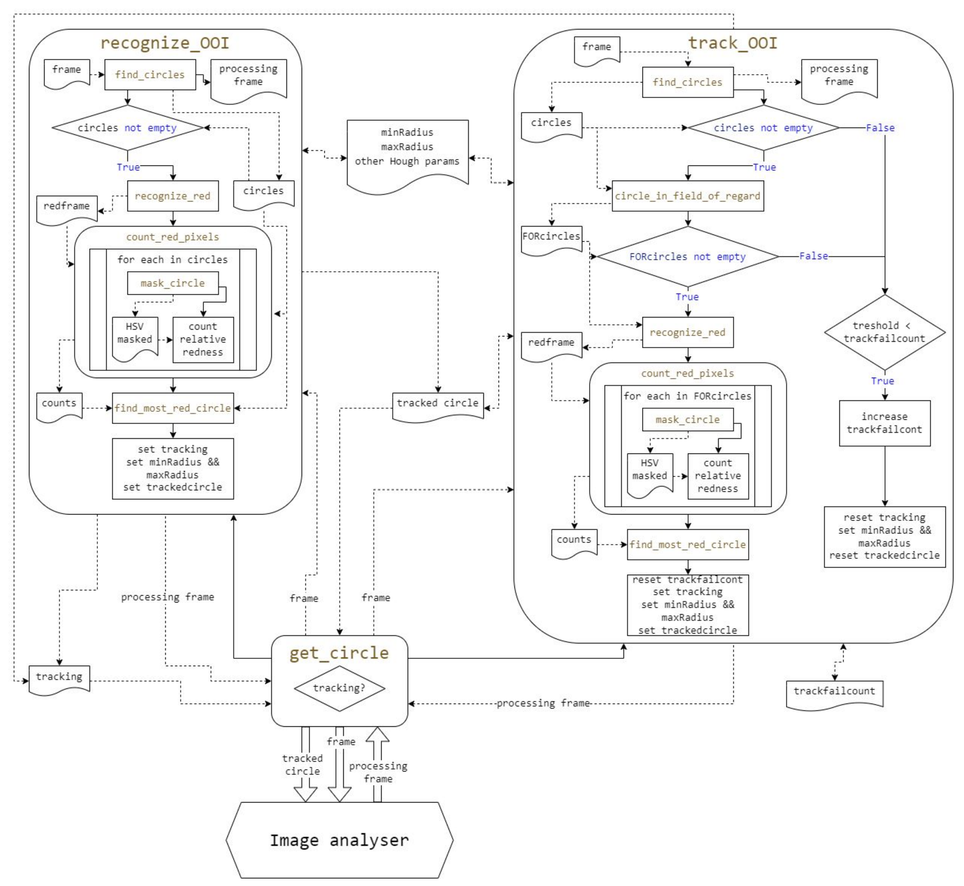
Main ideas behind the camera algorithm:

- We used Hough Circle Transform
- HSV instead of RGB to filter for red colour
- Separate methods to recognize and track the OOI
- Field Of Regard during tracking
- Adaptive expected circle radius

Challenges, possible solutions and improvements:

- Ball with current lighting conditions is hard for HCT
- Frequently losing track is hard to compensate for because of the slow update frequency
- Update frequency could be improved using FOR
- Switching to recognize red before circles?
- Use tracking to filter for radius
- Use tracking to predict movement?

CircleDetector class responsible for recognizing and tracking the OOI:

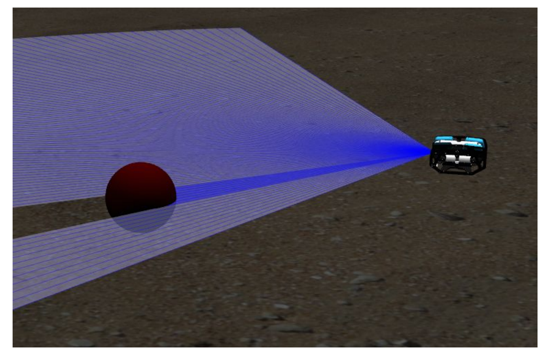


PERCEPTION: SONAR

- (1) ROS node to extract minimum range (closest point) from LaserMsg & publish OOI distance information
- (2) Implementation of existing "obstacle_detector" ROS package (by tysik) that takes LaserMsg as input & publishes CircleObstacle message (with x-y of centre & radius)

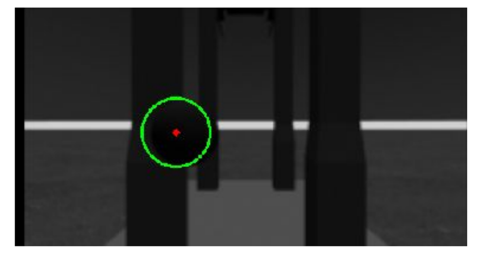
Future Work: Processing of real FLS data

- Backscatter 2D image analysis for circle identification (e.g. through adapted RANSAC algorithm) and/or
- Point cloud clustering and 3D object detection



OOI in Laser Scan range of ROV

Sonar laserscan object detection



Camera processed image object detection

CONTROL ALGORITHM

Design requirements

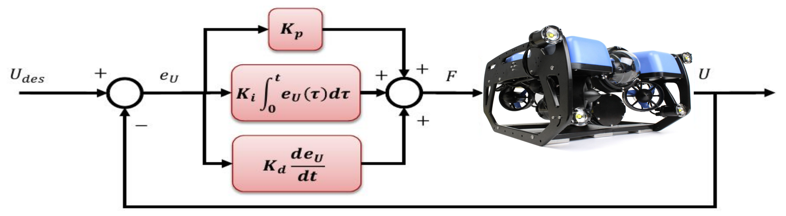
Position: Object (center) relative position

- X (surge): +/- 0.2m, based on camera and sonar resolution we must not be too far (max 5m) or close (min 1m)
- Y (yaw): +/- 10deg, should be safely within camera and sonar horizontal FOV 86.6 deg and 120 deg respectively.
- Z (heave): +/- 0.174m, `sin(10deg)*1m` - Sonar 20deg vertical aperture, min 1m distance

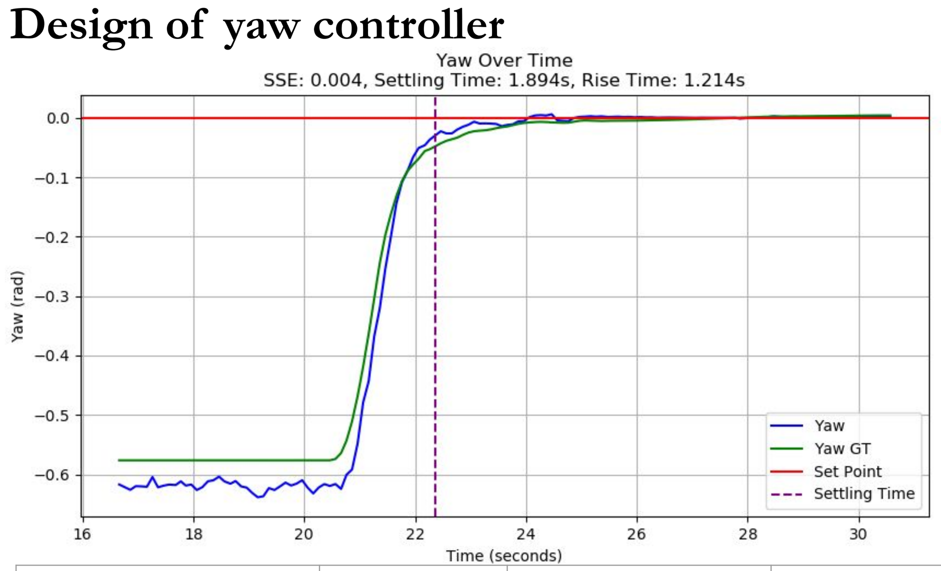
Velocity constraints

- Surge: maximum velocity of 1.5 m/s
- Heave: maximum velocity of 0.5 m/s

Single controller design



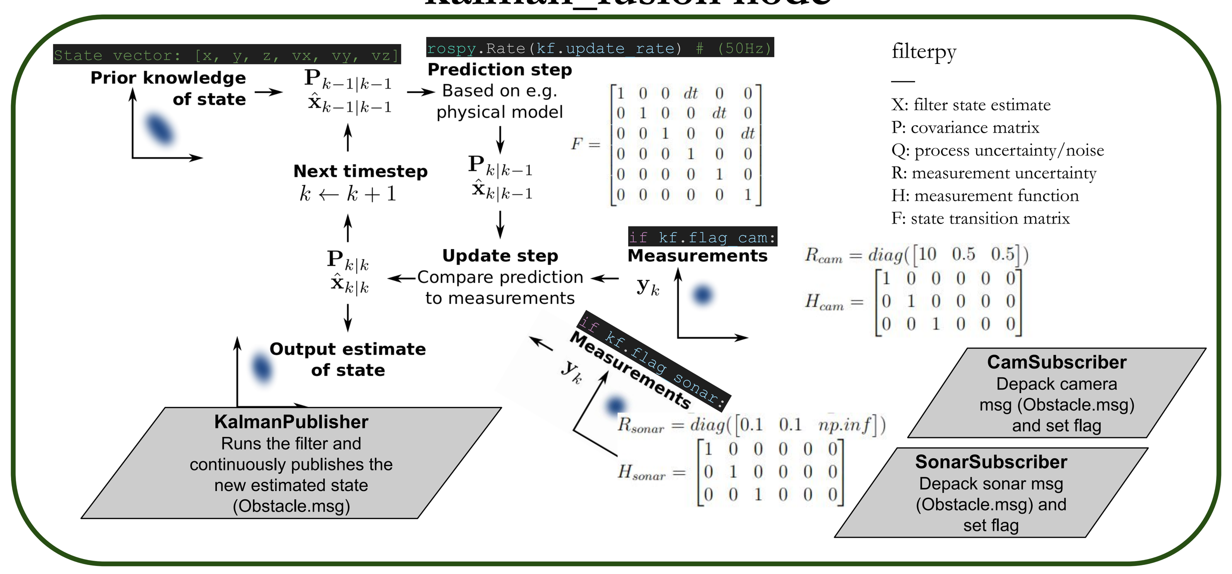
Design of yaw controller



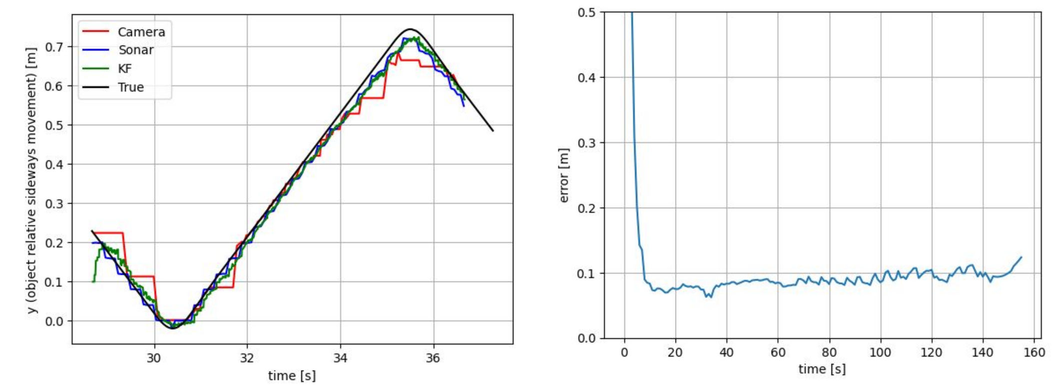
PID Controller performance	SSE [m]	Settling time [s]	Rise time [s]
Surge [3 m => 1 m]	0.021	5.002	3.096
Yaw [π/6 => 0]	0.004	1.894	1.214
Heave [0.5 m => 0 m]	0.004	2.920	3.698

SENSOR FUSION

kalman_fusion node

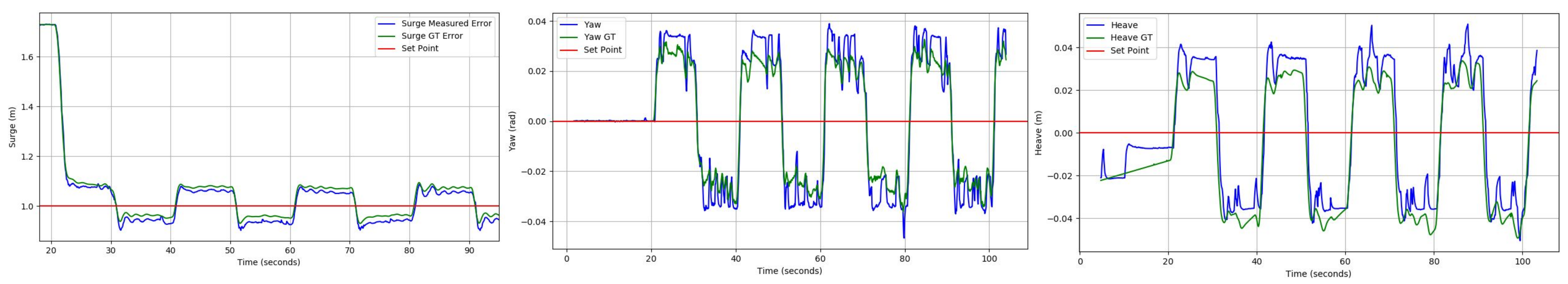


Verification of filter (test with horizontal movement)

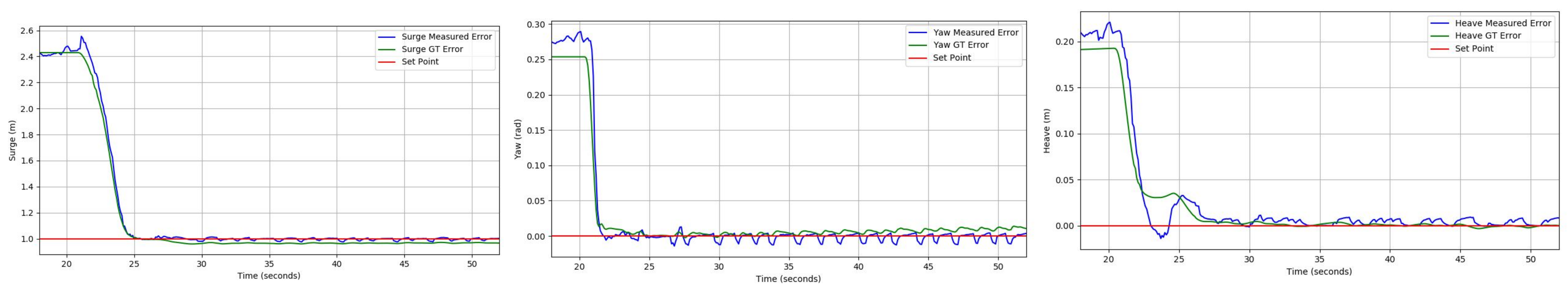


RESULTS - Simulation

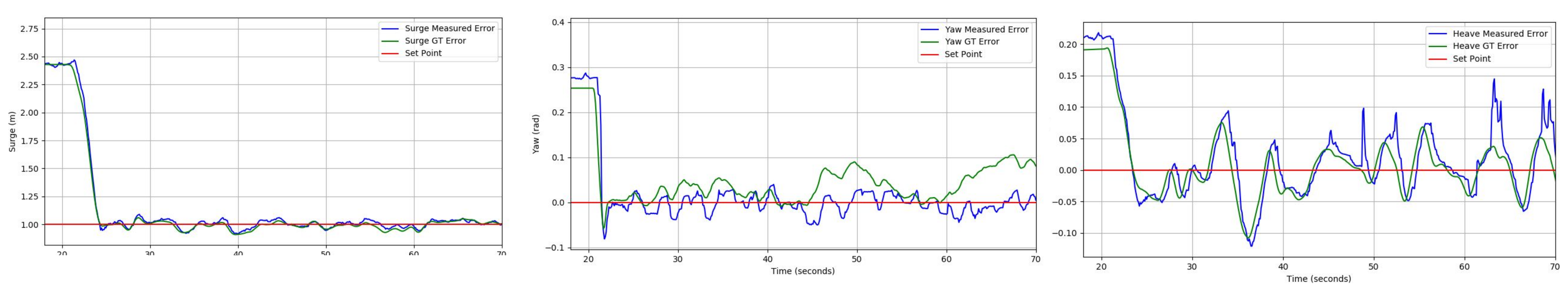
Experiment: Tracking



Experiment: 3-DoF setpoint



Experiment: Tracking + 3-DoF setpoint



CONCLUSION

Implementation of a pipeline detecting, tracking and following an object real time in simulation using ROS.

Fusing outputs from monocular camera and sonar (simulated as laserscan) using kalman filter.

Relative pose of the tracked object is processed by PID control algorithms sending surge, yaw and heave velocity commands to the ROV.

Next steps:
- Robustness in CV (e.g. multiple circles)
- Implement on actual ROV

REFERENCES

Whitt, C., Pearlman, J., Polagye, B., Caimi, F., Muller-Karger, F., Copping, A., ... & Khalsa, S. J. (2020). Future vision for autonomous ocean observations. Frontiers in Marine Science, 7, 697.

Einarsson, E. M., & Lipenitis, A. (2020). Model Predictive Control for the BlueROV2: Theory and Implementation (Master's thesis). Department of Energy Technology, Aalborg University, Esbjerg, Denmark